NAKED DATA vs UXP DATA TRANSFER

- 150 samples
- Client -> Linux Ubuntu
 - Memory: 8GBNo. of CPU: 4
 - o Intel(R) Core (TM) i3-7100U CPU @ 2.0 GHz
- The resource usage is measured by using Linux's inbuilt "TOP" command.
- The resource usage for naked byte includes transfer using naked socket whereas the resource consumption for UXP includes creation of the UXP plus the transfer of UXP using socket.

Table 1: Dataset Information

Naked Data	UXP object
1byte	25.3 kB (25,257 bytes)
1KB (1024bytes)	26.3kB (26,338 bytes)
10kB (10,000 bytes)	35.3kB (35,313 bytes)
100kB (100,000 bytes)	125.3kB (125,297 bytes)
500kB (500,000 bytes)	525.4 kB (525,421 bytes)
1MB (1,048,576 bytes)	1.1MB (1.074,149 bytes)

The following graph depicts Maximum CPU vs Payload.

- It shows that the CPU utilization is greater for UXP than Naked Byte Transfer.
- The differences of maximum CPU utilization between different payloads seems negligible, as they all seem to cluster around 70% to 80%.
- The CPU% is for single CPU, but since the client machine has 4 CPU, the actual CPU% should be divided by 4. (e.g., 80% means 80%/4 = 20% CPU usage)

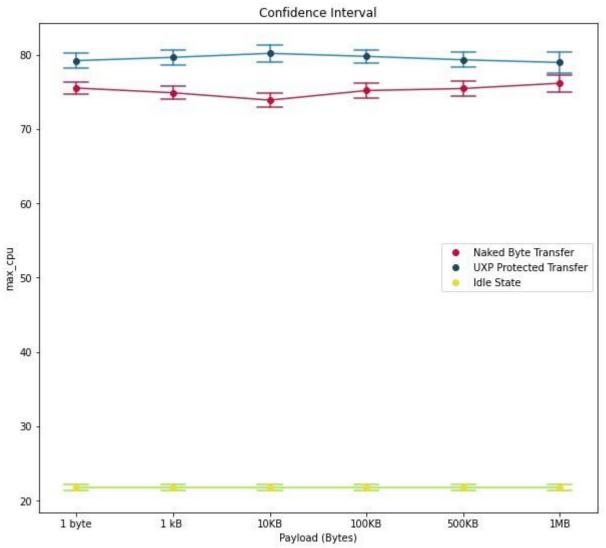


Figure 1: Max CPU vs Payload

The following graph depicts Maximum Memory vs Payload.

- It shows that the Memory utilization is greater for UXP than Naked Byte Transfer.
- The Memory usage seems to increase for higher sized UXP. But it seems constant for the naked byte transfer regardless of the size.
- This makes sense because creating UXP and transporting through socket should take higher memory than transporting naked data only.

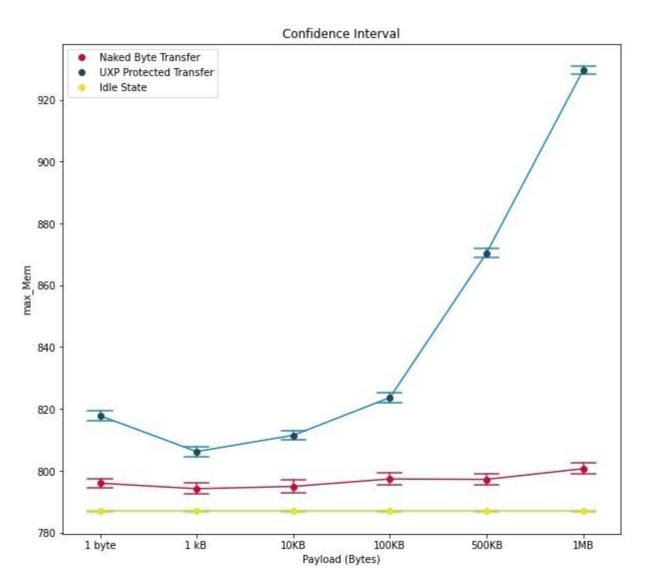


Figure 2: Max Memory vs Payload